<u>REMARKS</u>

Claims 1-10 and 12-21 remain in the present application. Claims 1-21 are rejected. Claim 11 is cancelled. Although Claims 1, 3, 10, 12, 14, and 19 are amended herein, Applicants respectfully submit that no new matter has been added by the claim amendments. Applicants respectfully request further examination and reconsideration of the rejections based on the amendments and arguments set forth below.

<u>Claim Objections</u>

Claims 2, 3, 13, and 14 are objected to in the above referenced Office Action with respect to the terms "memory" and "register." The examiner relies upon a definition from IEEE 100 The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition (2000), which Examiner cites as:

> **register (4)** A storage device or storage location having a specified storage capacity.

Applicants respectfully assert that this definition is unreasonably broad, as it includes devices clearly outside the scope of the present application (e.g., punch cards).

As such, Applicants respectfully submit that the following definition from Wikipedia (http://en.wikipedia.org/wiki/Processor_register) is a more

appropriate definition of the term "register" as it is commonly used today by those skilled in the art:

> In computer architecture, a processor register is a small amount of very fast computer memory used to speed the execution of computer programs by providing quick access to commonly used values— typically, the values being in the midst of a calculation at a given point in time. Most, but not all, modern computer architectures operate on the principle of moving data from main memory into registers, operating on them, then moving the result back into main memory—a so-called load-store architecture.

As stated in response to the prior Office Action, "memory" is a term referring generally to many different types of storage for a digital system, whereas "register" is a more specialized term generally referring to high-speed memory within a processor or other electronic device, used to hold data for a particular purpose. Thus, Applicants respectfully assert that "memory" and "register" are not synonymous, thereby rendering distinguishable Claims 2 and 3, and Claims 13 and 14.

Moreover, to further emphasize this distinction, Claims 3 and 14 have been amended. Applicants respectfully assert that no new matter has been added as a result of these claim amendments, and that these Claim amendments are adequately supported by the application and figures as originally filed. Consequently, in light of these claim amendments and the discussion above, Applicants respectfully request that all objections relating to the use of the terms "memory" and "register" be withdrawn.

Claim Rejections - 35 U.S.C. §112

Claims 12 and 19 have been amended to obviate the cited 35 U.S.C. §112 rejections.

Claim Rejections - 35 U.S.C. §103

The above referenced Office Action rejects Claims 1-21 as being unpatentable over U.S. Patent No. 6,202,044 (hereafter referred to as "Tzori"). Applicants respectfully traverse.

Applicants respectfully direct the Examiner to independent Claim 1 that recites a boot method for an In-Circuit Emulation system having a microcontroller operating in lock-step synchronization with a virtual microcontroller, comprising (emphasis added):

> <u>in the microcontroller, executing a set of boot code;</u>
> in the virtual microcontroller, <u>executing a set of timing code to enable the lock-step synchronization, wherein the timing code is timed to take the same number of clock cycles as the microcontroller uses to execute the boot code</u>, and wherein the boot code is stored within the microcontroller and at least one portion of the boot code is inaccessible to the virtual microcontroller; and
> simultaneously halting both the microcontroller and the virtual microcontroller.

Independent Claims 10 and 12 recite limitations similar to those in Claim 1.

Claims 2-9 depend from independent Claim 1 and recite further limitations

to the claimed invention. Claims 13-21 depend from independent Claim 12 and recite further limitations to the claimed invention.

Applicants respectfully assert that Tzori fails to suggest, teach, or describe the limitation "in the microcontroller, executing a set of boot code" as recited in Claim 1. Embodiments of the claimed invention disclose boot process for in In-Circuit Emulator system (page 4, lines 28-29). More specifically, the boot process involves running boot code in the microcontroller (page 5, lines 2-3).

In contrast, Applicants understand Tzori to teach an initialization interval that involves loading logic configuration data into configurable-logic ICs within the hardware pod as illustrated in Figure 2 (col. 9, lines 20-27). After the initialization interval, Tzori goes on to teach that the simulation cycles begin immediately thereafter (col. 9, lines 27-30). As such, assuming arguendo that the microcontroller recited in Claim 1 is equivalent to device under test inserted into IC socket 34 as shown in Figure 2 of Tzori, Tzori is very different from the claimed invention. Applicants respectfully assert that whereas the claimed invention involves <u>running boot code in the microcontroller</u> amongst other operations to enable lock-step synchronization (line 28 of page 4 to line 9 of page 5), Tzori teaches <u>loading logic configuration data into devices other than the device under test</u> during the initialization

interval. Thus, not only is the microcontroller as claimed very different from the configurable-logic ICs taught by Tzori, but the operations described (running code vs. loading data into) and the information involved (boot code vs. logic configuration data) are also very different. Moreover, by teaching the loading of logic configuration data into devices other than the device under test to initialize the simulator, Tzori effectively <u>teaches away</u> from the claimed embodiments.

Applicants respectfully assert that Tzori fails to suggest, teach, or describe the limitation "executing a set of timing code to enable the lock-step synchronization, wherein the timing code is timed to take the same number of clock cycles as the microcontroller uses to execute the boot code" as recited in Claim 1. Embodiments of the claimed invention disclose running timing code in the virtual microcontroller while the microcontroller simultaneously runs boot code to enable lock-step synchronization between the microcontroller and the virtual microcontroller (page 5, lines 2-9).

In contrast, Applicants understand Tzori to teach loading logic configuration data into configurable-logic ICs during the initialization interval as discussed above. As such, assuming arguendo that the simulation process and server process taught in Tzori are equivalent to the virtual microcontroller as claimed, Applicants respectfully assert that Tzori <u>teaches</u>

<u>away</u> from running timing code as claimed given that Tzori instead teaches that the simulation process loads logic configuration data during the initialization interval.

For these reasons, Applicants respectfully assert that independent Claims 1, 10, and 12 are not rendered obvious by Tzori. Since Claims 2-9 depend from independent Claim 1, and Claims 13-21 depend from independent Claim 12, the dependent Claims are also not rendered obvious by Tzori. Thus, Claims 1-10 and 12-21 overcome the 35 U.S.C. §103(a) rejections of record, leaving these claims in a state of allowance.

CONCLUSION

Applicants respectfully assert that Claims 1-10 and 12-21 are in condition for allowance and Applicants earnestly solicit such action from the Examiner.
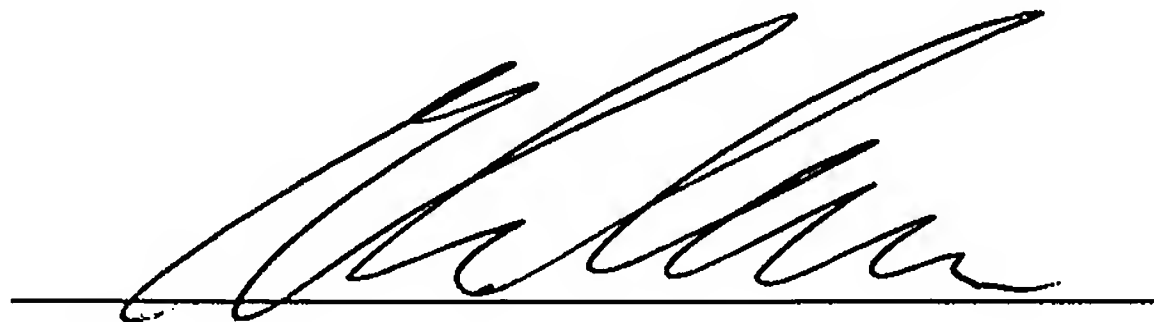
The Examiner is urged to contact Applicants' undersigned representative if the Examiner believes such action would expedite resolution of the present Application.

Please charge any additional fees or apply any credits to our PTO deposit account number: 23-0085.

Respectfully submitted,

WAGNER, MURABITO & HAO, LLP

Dated: 12/22, 2005

Glenn Barnes
Registration No. 42,293

Two North Market Street
Third Floor
San Jose, CA 95113
(408) 938-9060